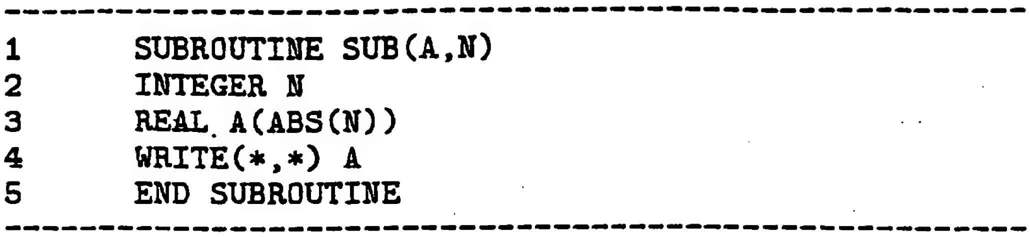
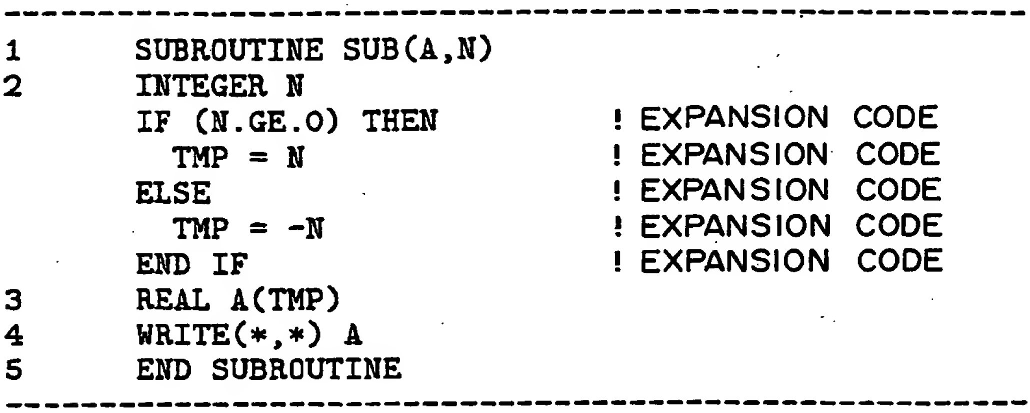


```
1       SUBROUTINE SUB(A,N)
2       INTEGER N
3       REAL A(ABS(N))
4       WRITE(*,*) A
5       END SUBROUTINE
```

FIG. 1A



```
1       SUBROUTINE SUB(A,N)
2       INTEGER N
        IF (N.GE.0) THEN            ! EXPANSION CODE
          TMP = N                   ! EXPANSION CODE
        ELSE                        ! EXPANSION CODE
          TMP = -N                  ! EXPANSION CODE
        END IF                      ! EXPANSION CODE
3       REAL A(TMP)
4       WRITE(*,*) A
5       END SUBROUTINE
```

FIG. 1B

```
char *copy_string(char *s)
{
 int i;
 char *buffer = (char*)malloc(strlen(s) + 1);

 for (i = 0; s[i] != '\0'; ++i)
  buffer[i] = s[i];

 return buffer;
}
```

FIG. 2A

```
   char *copy_string(char *s)
   {
    int i;
char *p; /* EXPANSION CODE */
int tmp; /* EXPANSION CODE */
tmp = 0; /* EXPANSION CODE */
for (p = s; *p != '\0'; ++p) /*EXPANSION CODE */
 ++tmp; /* EXPANSION CODE */
    char *buffer = (char*)malloc(tmp + 1);

    for (i = 0; s[i] != '\0'; ++i)
     buffer[i] = s[i];

    return buffer;
   }
```

FIG. 2B

```
------------------------------------------------------------
1       IF (Z.GT.EPS) THEN
2         A=B1
3       ELSE IF(ABS(Z).LE.EPS) THEN
4         A=B2
5       ELSE
6         A=B3
7       END IF
------------------------------------------------------------
```

FIG. 3A

```
------------------------------------------------------------
1       IF (Z.GT.EPS) THEN
2         A=B1
3a      ELSE
          IF (Z.GE.0.0) THEN        ! EXPANSION  CODE
            TMP = Z                 ! EXPANSION  CODE
          ELSE                      ! EXPANSION  CODE
            TMP = -Z                ! EXPANSION  CODE
          END IF                    ! EXPANSION  CODE
3b        IF(TMP.LE.EPS) THEN
4           A=B2
5         ELSE
6           A=B3
3c        END IF
7       END IF
------------------------------------------------------------
```

FIG. 3B

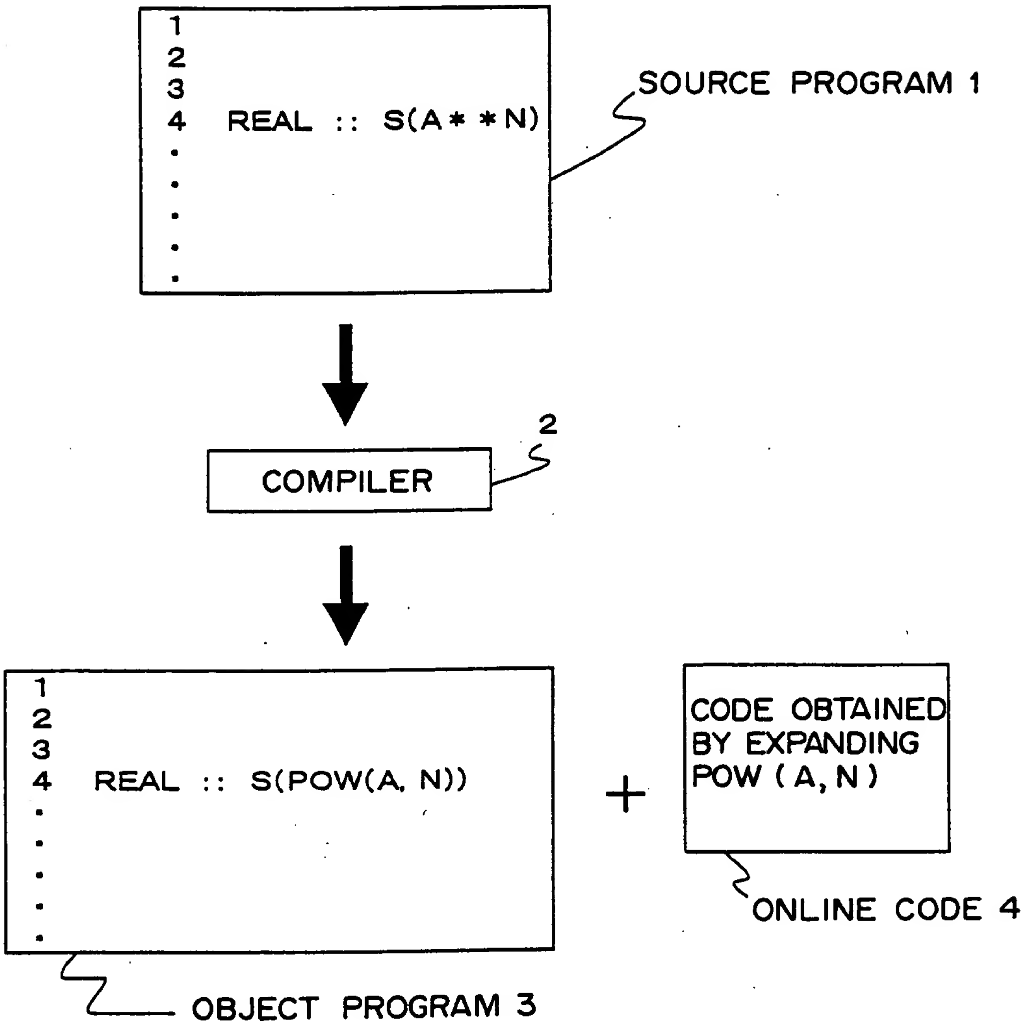

1
2
3
4 REAL :: S(A**N)
SOURCE PROGRAM 1
COMPILER
2
1
2
3
4 REAL :: S(POW(A, N))
OBJECT PROGRAM 3
+
CODE OBTAINED BY EXPANDING POW (A, N)
ONLINE CODE 4

FIG. 4

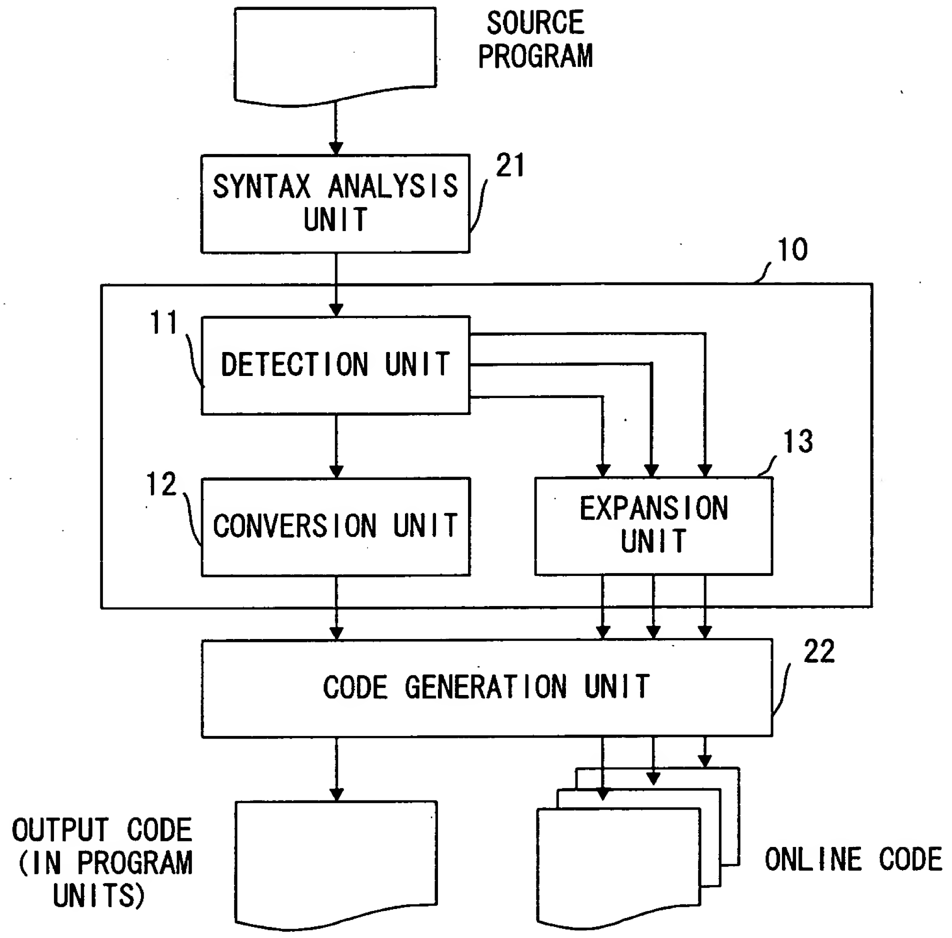

SOURCE PROGRAM
SYNTAX ANALYSIS UNIT
21
10
11
DETECTION UNIT
12
CONVERSION UNIT
13
EXPANSION UNIT
CODE GENERATION UNIT
22
OUTPUT CODE (IN PROGRAM UNITS)
ONLINE CODE

FIG. 5

INPUT: PROGRAM UNIT P
OUTPUT: P' OBTAINED BY AMENDING P, AND PROCEDURE S1,…, Sn (0≦n)

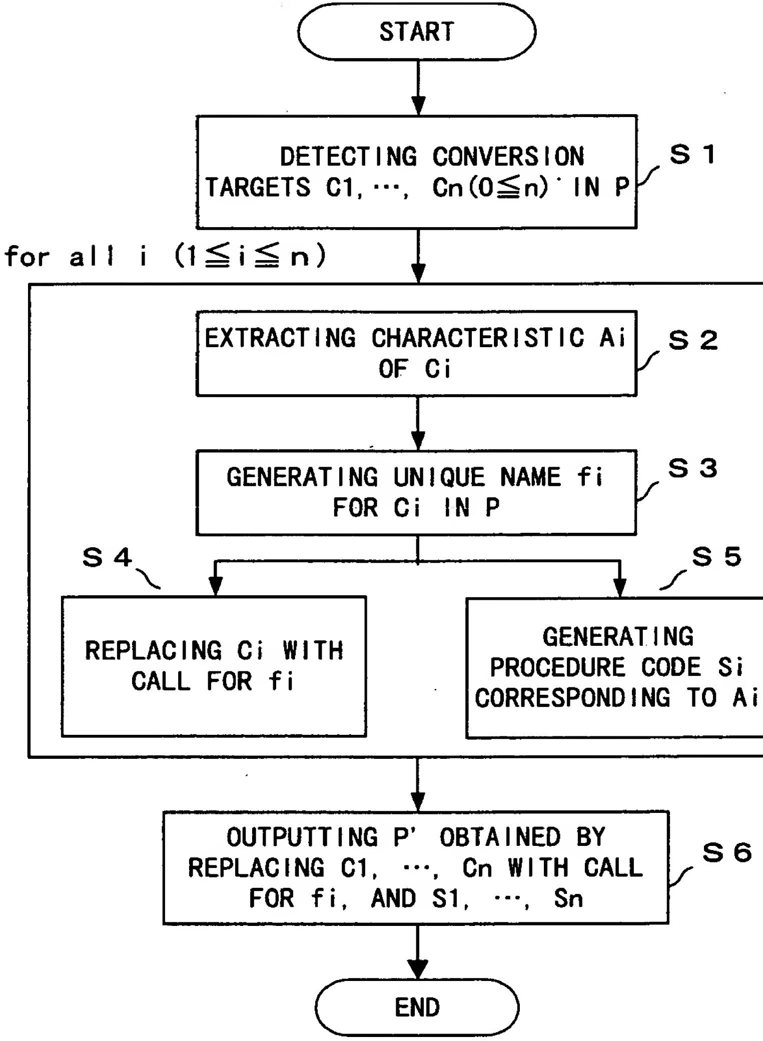

START
DETECTING CONVERSION TARGETS C1,…, Cn(0≦n) IN P
S1
for all i (1≦i≦n)
EXTRACTING CHARACTERISTIC Ai OF Ci
S2
GENERATING UNIQUE NAME fi FOR Ci IN P
S3
S4
S5
REPLACING Ci WITH CALL FOR fi
GENERATING PROCEDURE CODE Si CORRESPONDING TO Ai
OUTPUTTING P' OBTAINED BY REPLACING C1, …, Cn WITH CALL FOR fi, AND S1, …, Sn
S6
END

FIG. 6

```
1       PROGRAM SAMPL
2       INTEGER N(100)
3       REAL A(10,20,30),B
        ...
4       B = SUM(A)
5       WRITE(*,*) SUM(N(51:100))
6       END
```

FIG. 7A

```
1       PROGRAM SAMPL
2       INTEGER N(100)
3       REAL A(10,20,30),B
        ...
4       B = SUM_SAMPL_1(A)
5       WRITE(*,*) SUM_SAMPL_2(N(51:100))
6       END
```

FIG. 7B

```
      arg-type FUNCTION SUM(X)
      arg-type X(lb(1):ub(1), ···, lb(m):ub(m))
      SUM = 0
      DO 999 Im = lb(m), ub(m)
        ⋮
      DO 999 I1 = lb(1), ub(1)
       SUM = SUM+X(I1,···,Im)
999   CONTINUE
      RETURN
      END
```

FIG. 8

```
      REAL FUNCTION SUM_SAMPL_1(X)
      REAL X(1:10,1:20,1:30)
      SUM_SAMPL_1 = 0
      DO 999 I3 = 1, 30
      DO 999 I2 = 1, 20
      DO 999 I1 = 1, 10
       SUM_SAMPL_1 = SUM_SAMPL_1+X(I1,I2,I3)
  999 CONTINUE
      RETURN
      END
```

FIG. 9A

```
      INTEGER FUNCTION SUM_SAMPL_2(X)
      INTEGER X(51:100)
      SUM_SAMPL_2 = 0
      DO 999 I1 = 51, 100
       SUM_SAMPL_2 = SUM_SAMPL_2+X(I1)
  999 CONTINUE
      RETURN
      END
```

FIG. 9B

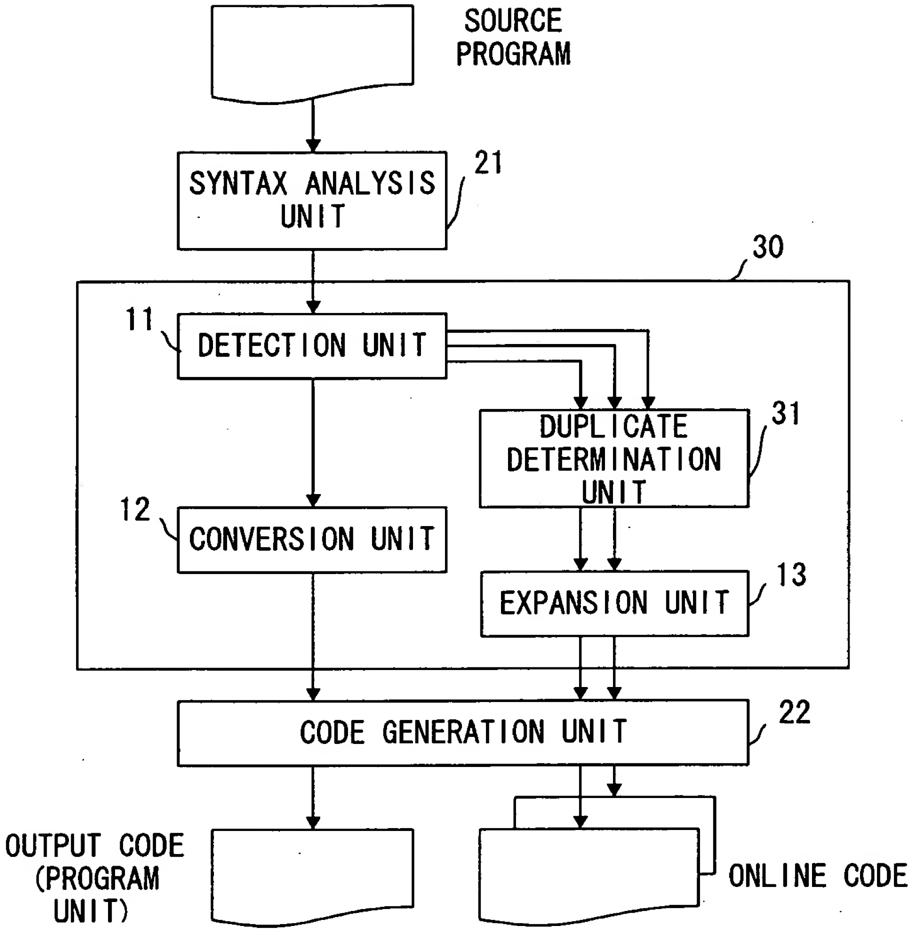

SOURCE PROGRAM
SYNTAX ANALYSIS UNIT
21
30
11
DETECTION UNIT
DUPLICATE DETERMINATION UNIT
31
12
CONVERSION UNIT
EXPANSION UNIT
13
CODE GENERATION UNIT
22
OUTPUT CODE (PROGRAM UNIT)
ONLINE CODE

FIG. 10

INPUT: PROGRAM UNIT P
OUTPUT: P' OBTAINED BY AMENDING P, AND PROCEDURE S1, . . . , Sm(0≦m≦n)

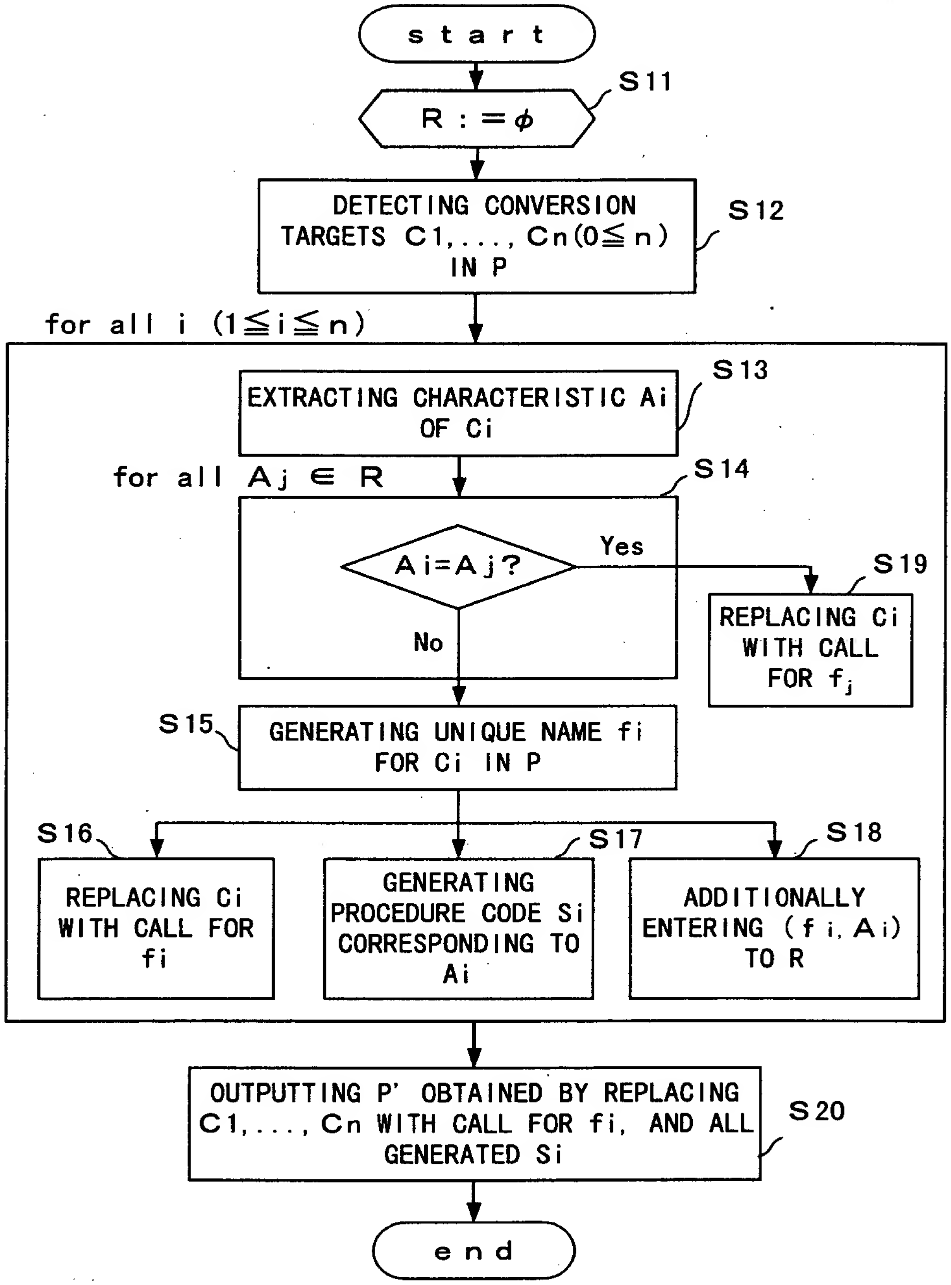

start
S11
R := φ
DETECTING CONVERSION TARGETS C1, . . . , Cn(0≦n) IN P
S12
for all i (1≦i≦n)
EXTRACTING CHARACTERISTIC Ai OF Ci
S13
for all Aj ∈ R
S14
Ai=Aj?
Yes
No
S19
REPLACING Ci WITH CALL FOR fj
S15
GENERATING UNIQUE NAME fi FOR Ci IN P
S16
REPLACING Ci WITH CALL FOR fi
S17
GENERATING PROCEDURE CODE Si CORRESPONDING TO Ai
S18
ADDITIONALLY ENTERING (fi, Ai) TO R
OUTPUTTING P' OBTAINED BY REPLACING C1, . . . , Cn WITH CALL FOR fi, AND ALL GENERATED Si
S20
end

FIG. 11

```
      PROGRAM SAMPL
      INTEGER N(100),M(200)
      REAL A(10,20,30),A2(10,20,30),B
      ...
      B = SUM(A)+SUM(A2)
      WRITE(*,*) SUM(N(51:100))
      WRITE(*,*) SUM(M(51:200))
      END
```

FIG. 12A

```
      PROGRAM SAMPL
      INTEGER N(100),M(200)
      REAL A(10,20,30),A2(10,20,30),B
      ...
      B = SUM_SAMPL_1(A)+SUM_SAMPL_1(A2)
      WRITE(*,*) SUM_SAMPL_2(N(51:100))
      WRITE(*,*) SUM_SAMPL_3(M(51:200))
      END
```

FIG. 12B

CALL	arg−type	m	lb(1)	ub(1)	lb(2)	ub(2)	lb(3)	ub(3)
SUM__SAMPL__1	REAL	3	1	10	1	20	1	30

FIG. 13A

CALL	arg−type	m	lb(1)	ub(1)	lb(2)	ub(2)	lb(3)	ub(3)
SUM__SAMPL__1	REAL	3	1	10	1	20	1	30
NEWLY EXTRACTED CALL	REAL	3	1	10	1	20	1	30

FIG. 13B

CALL	arg−type	m	lb(1)	ub(1)	lb(2)	ub(2)	lb(3)	ub(3)
SUM__SAMPL__1	REAL	3	1	10	1	20	1	30
NEWLY EXTRACTED CALL	INTEGER	1	51	100	—	—	—	—

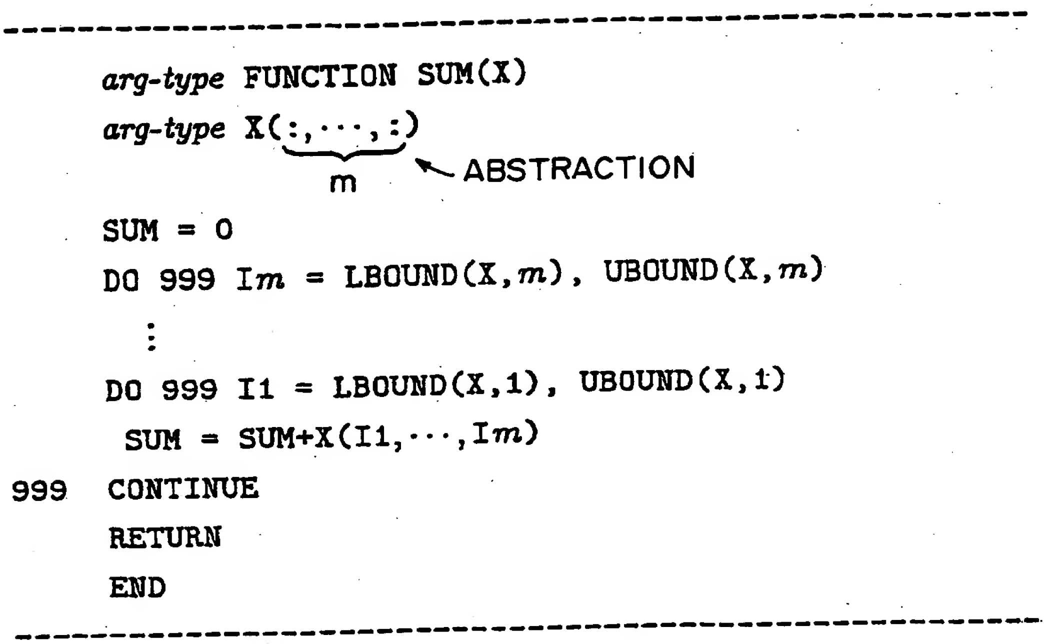
FIG. 13C

CALL	arg−type	m	lb(1)	ub(1)	lb(2)	ub(2)	lb(3)	ub(3)
SUM__SAMPL__1	REAL	3	1	10	1	20	1	30
SUM__SAMPL__2	INTEGER	1	51	100	—	—	—	—
NEWLY EXTRACTED CALL	INTEGER	1	51	200	—	—	—	—

FIG. 13D

```
      INTEGER FUNCTION SUM_SAMPL_3(X)
      INTEGER X(51:200)
      SUM_SAMPL_3 = 0
      DO 999 I1 = 51, 200
       SUM_SAMPL_3 = SUM_SAMPL_3+X(I1)
  999 CONTINUE
      RETURN
      END
```

FIG. 14



```
      arg-type FUNCTION SUM(X)
      arg-type X(:,···,:)
      SUM = 0
      DO 999 Im = LBOUND(X,m), UBOUND(X,m)
        :
      DO 999 I1 = LBOUND(X,1), UBOUND(X,1)
       SUM = SUM+X(I1,···,Im)
999   CONTINUE
      RETURN
      END
```

m
ABSTRACTION

FIG. 15

CALL	*arg−type*	*m*
SUM(A)	REAL	3
SUM(A2)	REAL	3
SUM(N(51:100))	INTEGER	1
SUM(M(51:200))	INTEGER	1

FIG. 16

```
     PROGRAM SAMPL
     INTEGER N(100),M(200)
     REAL A(10,20,30),A2(10,20,30),B
     ...
     B = SUM_SAMPL_1(A)+SUM_SAMPL_1(A2)
     WRITE(*,*) SUM_SAMPL_2(N(51:100))
     WRITE(*,*) SUM_SAMPL_2(M(51:200))
     END
```

OBJECT CODE

```
     REAL FUNCTION SUM_SAMPL_1(X)
     REAL X(:,:,:)
     SUM_SAMPL_1 = 0
     DO 999 I3 = LBOUND(X,3),UBOUND(X,3)
     DO 999 I2 = LBOUND(X,2),UBOUND(X,2)
     DO 999 I1 = LBOUND(X,1),UBOUND(X,1)
      SUM_SAMPL_1 = SUM_SAMPL_1+X(I1,I2,I3)
 999 CONTINUE
     RETURN
     END
```

PROCEDURE CODE A

```
     INTEGER FUNCTION SUM_SAMPL_2(X)
     INTEGER X(:)
     SUM_SAMPL_2 = 0
     DO 999 I1 = LBOUND(X,1),UBOUND(X,1)
      SUM_SAMPL_2 = SUM_SAMPL_2+X(I1)
 999 CONTINUE
     RETURN
     END
```

PROCEDURE CODE B

FIG. 17

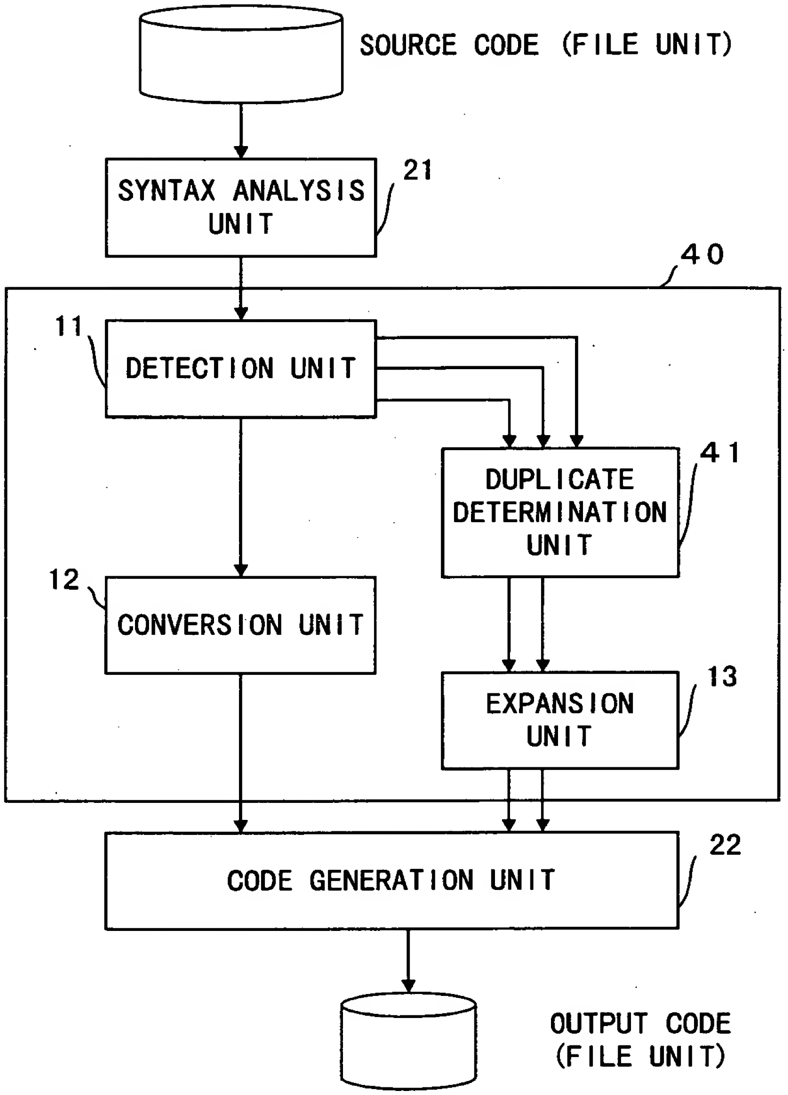
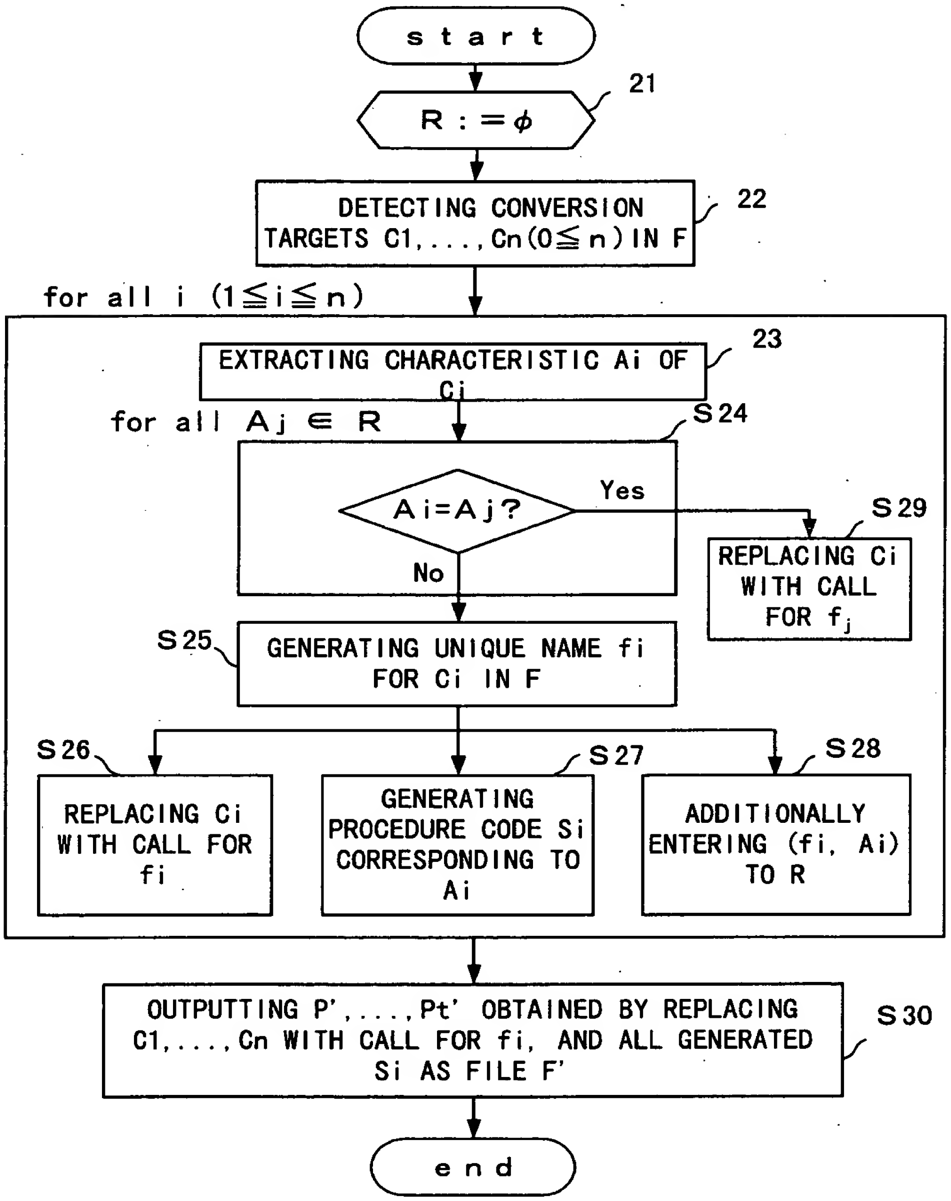

SOURCE CODE (FILE UNIT)
SYNTAX ANALYSIS UNIT
21
40
11
DETECTION UNIT
DUPLICATE DETERMINATION UNIT
41
12
CONVERSION UNIT
EXPANSION UNIT
13
CODE GENERATION UNIT
22
OUTPUT CODE (FILE UNIT)

FIG. 18

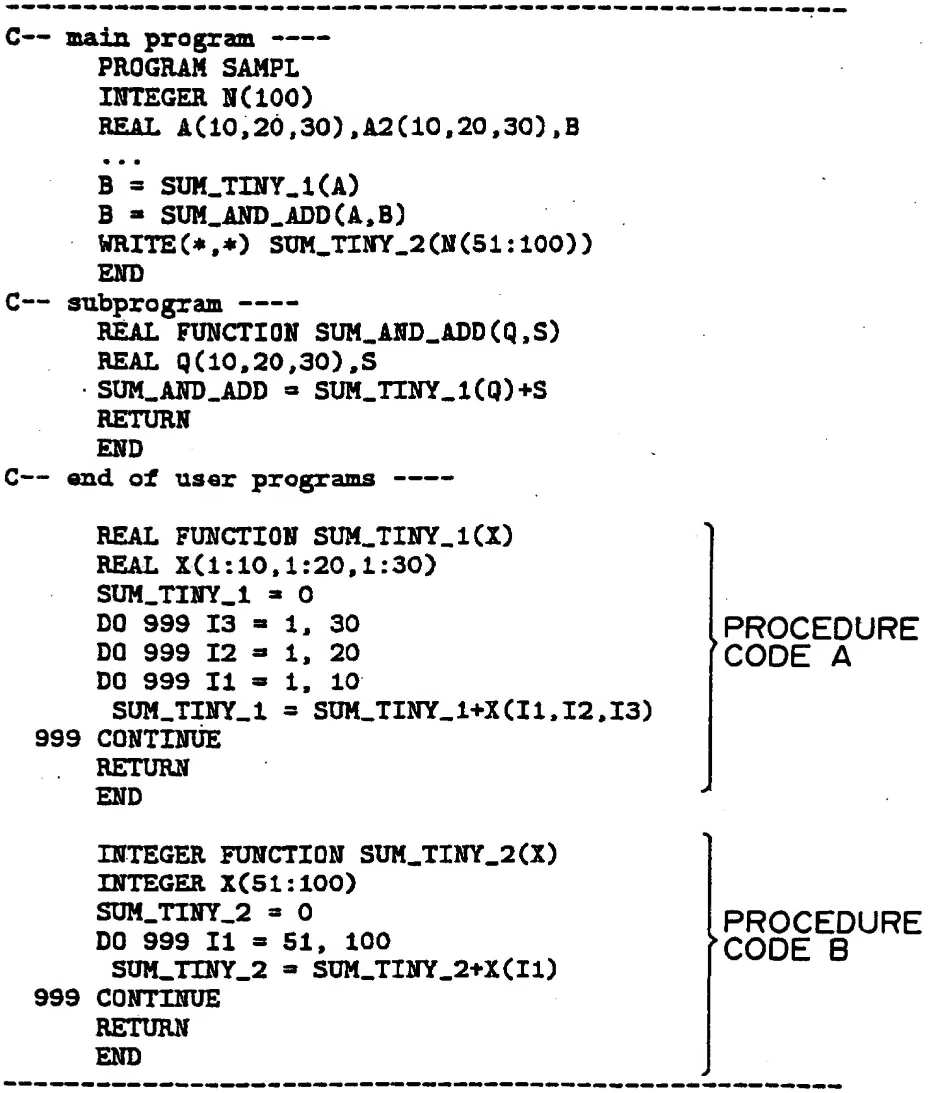
INPUT : FILE F CONTAINING PROGRAM UNIT P1, . . . , Pt (1≦t)
OUTPUT : FILE F' CONTAINING P', . . . , Pt' OBTAINED BY AMENDING P1', ···, Pt',
AND PROCEDURE S1, . . . , Sm (0≦m≦n)


start
21
R : = φ
22
DETECTING CONVERSION TARGETS C1, . . . , Cn (0≦n) IN F
for all i (1≦i≦n)
23
EXTRACTING CHARACTERISTIC Ai OF Ci
for all Aj ∈ R
S24
Ai=Aj?
Yes
No
S29
REPLACING Ci WITH CALL FOR fj
S25
GENERATING UNIQUE NAME fi FOR Ci IN F
S26
REPLACING Ci WITH CALL FOR fi
S27
GENERATING PROCEDURE CODE Si CORRESPONDING TO Ai
S28
ADDITIONALLY ENTERING (fi, Ai) TO R
OUTPUTTING P', . . . , Pt' OBTAINED BY REPLACING C1, . . . , Cn WITH CALL FOR fi, AND ALL GENERATED Si AS FILE F'
S30
end

F I G. 19

```
------------------------------------------------------------
C-- main program ----
      PROGRAM SAMPL
      INTEGER N(100)
      REAL A(10,20,30),A2(10,20,30),B
      ...
      B = SUM(A)
      B = SUM_AND_ADD(A,B)
      WRITE(*,*) SUM(N(51:100))
      END
C-- subprogram ----
      REAL FUNCTION SUM_AND_ADD(Q,S)
      REAL Q(10,20,30),S
      SUM_AND_ADD = SUM(Q)+S
      RETURN
      END
C-- end of user programs ----
------------------------------------------------------------
```

FIG. 20

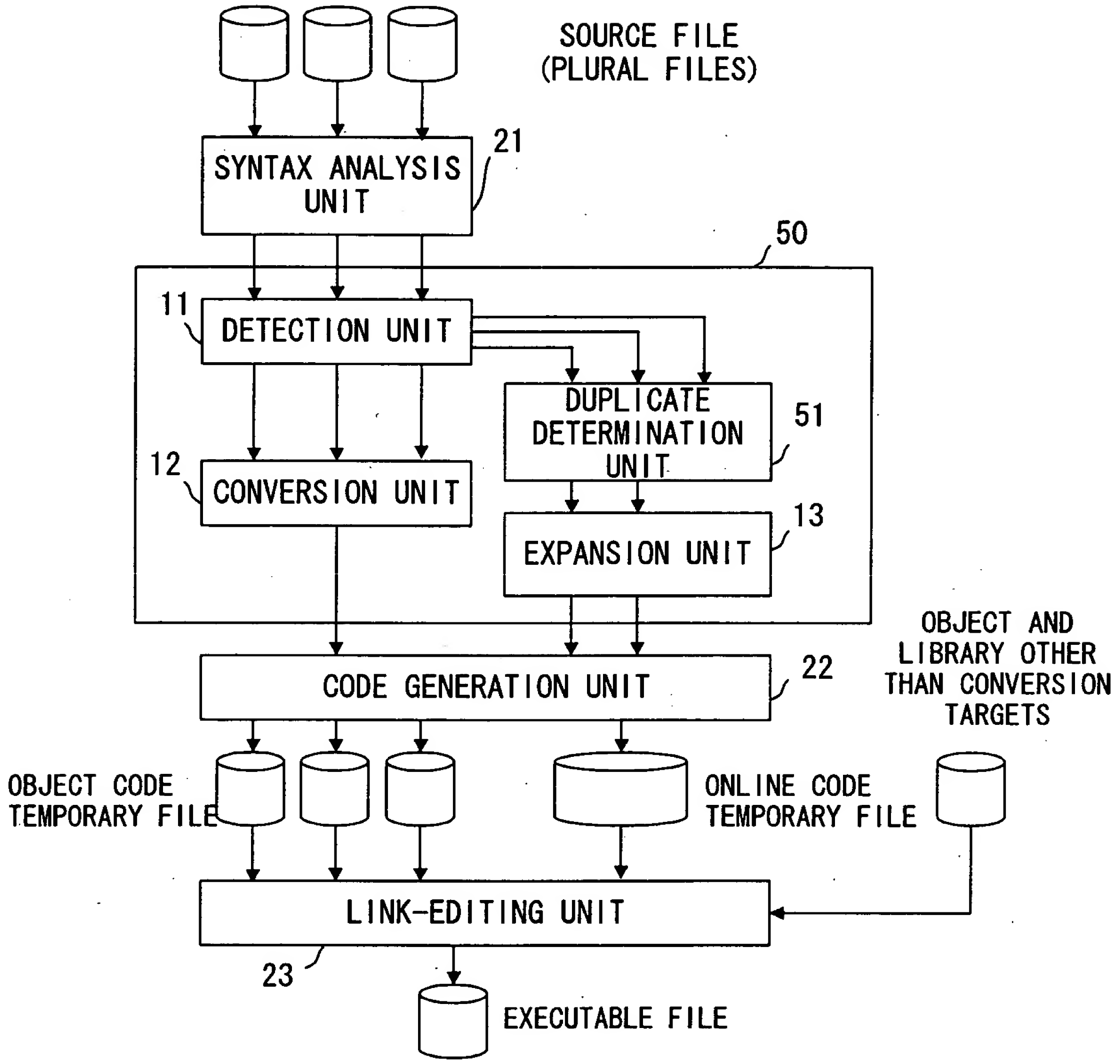


```
C-- main program ----
      PROGRAM SAMPL
      INTEGER N(100)
      REAL A(10,20,30),A2(10,20,30),B
      ...
      B = SUM_TINY_1(A)
      B = SUM_AND_ADD(A,B)
      WRITE(*,*) SUM_TINY_2(N(51:100))
      END
C-- subprogram ----
      REAL FUNCTION SUM_AND_ADD(Q,S)
      REAL Q(10,20,30),S
      SUM_AND_ADD = SUM_TINY_1(Q)+S
      RETURN
      END
C-- end of user programs ----

      REAL FUNCTION SUM_TINY_1(X)
      REAL X(1:10,1:20,1:30)
      SUM_TINY_1 = 0
      DO 999 I3 = 1, 30
      DO 999 I2 = 1, 20
      DO 999 I1 = 1, 10
       SUM_TINY_1 = SUM_TINY_1+X(I1,I2,I3)
  999 CONTINUE
      RETURN
      END

      INTEGER FUNCTION SUM_TINY_2(X)
      INTEGER X(51:100)
      SUM_TINY_2 = 0
      DO 999 I1 = 51, 100
       SUM_TINY_2 = SUM_TINY_2+X(I1)
  999 CONTINUE
      RETURN
      END
```

FIG. 21


SOURCE FILE
(PLURAL FILES)
SYNTAX ANALYSIS UNIT
21
50
11
DETECTION UNIT
DUPLICATE DETERMINATION UNIT
51
12
CONVERSION UNIT
13
EXPANSION UNIT
OBJECT AND LIBRARY OTHER THAN CONVERSION TARGETS
CODE GENERATION UNIT
22
OBJECT CODE TEMPORARY FILE
ONLINE CODE TEMPORARY FILE
LINK-EDITING UNIT
23
EXECUTABLE FILE

F I G. 2 2

INPUT : FILES F1, . . . , Fs (1≦s) CONTAINING PROGRAM UNITS P1, . . . , Pt (1≦t)
OUTPUT : FILE FO CONTAINING F1', . . . , Fs' OBTAINED BY AMENDING
F1, . . . , Fs, AND PROCEDURES S1, . . . , Sm (0≦m≦n)

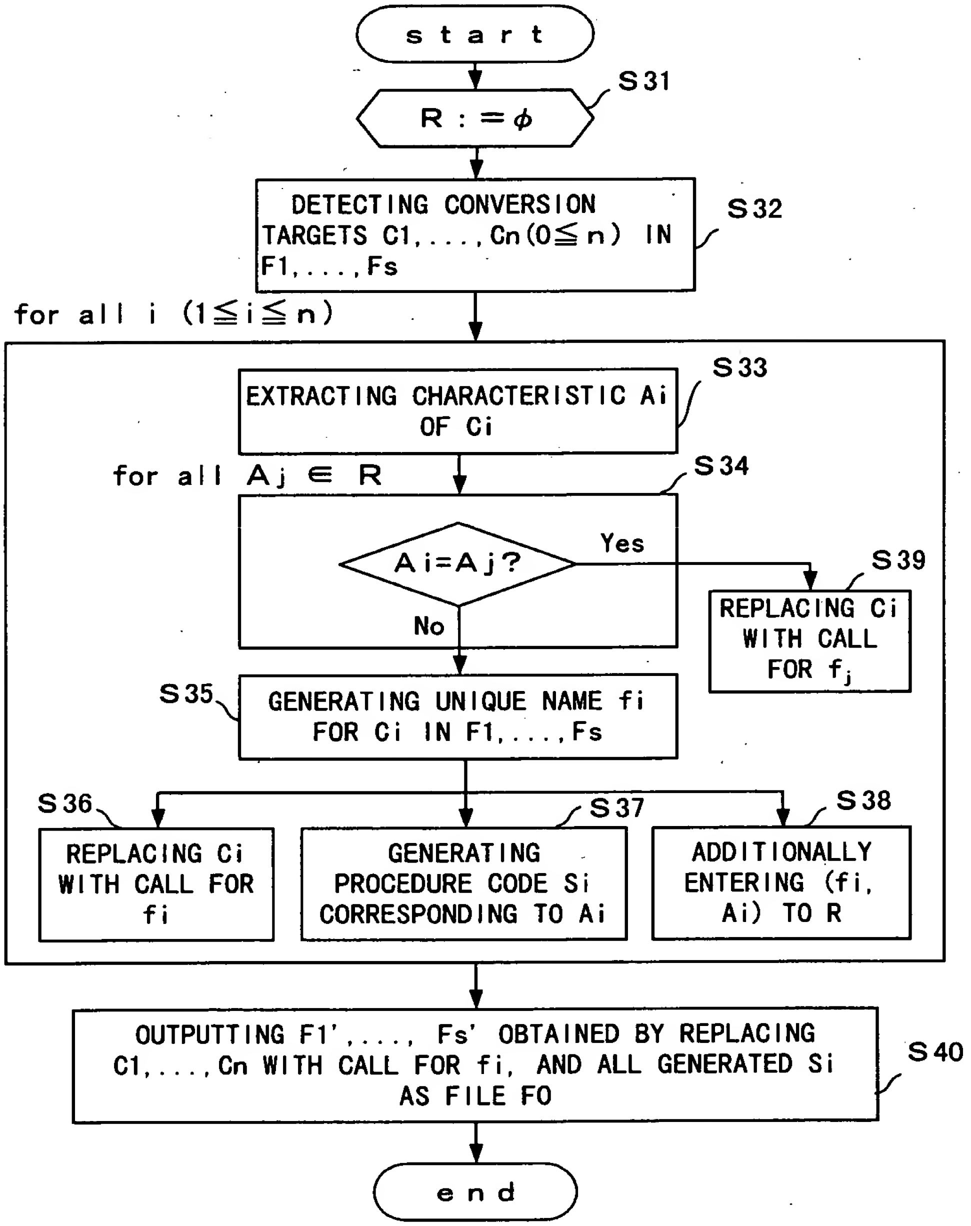

start
S31
R : =φ
DETECTING CONVERSION TARGETS C1, . . . , Cn (0≦n) IN F1, . . . , Fs
S32
for all i (1≦i≦n)
EXTRACTING CHARACTERISTIC Ai OF Ci
S33
for all Aj ∈ R
S34
Ai=Aj?
Yes
No
S39
REPLACING Ci WITH CALL FOR fj
S35
GENERATING UNIQUE NAME fi FOR Ci IN F1, . . . , Fs
S36
REPLACING Ci WITH CALL FOR fi
S37
GENERATING PROCEDURE CODE Si CORRESPONDING TO Ai
S38
ADDITIONALLY ENTERING (fi, Ai) TO R
OUTPUTTING F1', . . . , Fs' OBTAINED BY REPLACING C1, . . . , Cn WITH CALL FOR fi, AND ALL GENERATED Si AS FILE FO
S40
end

FIG. 23

```
  FILE tiny1.f:
------------------------------------------------------------
C-- main program ----
      PROGRAM SAMPL
      INTEGER N(100)
      REAL A(10,20,30),A2(10,20,30),B
      ...
      B = SUM(A)
      B = SUM_AND_ADD(A,B)
      WRITE(*,*) SUM(N(51:100))
      END
C-- end of main program ----
------------------------------------------------------------
  FILE tiny2.f:
------------------------------------------------------------
C-- subprogram ----
      REAL FUNCTION SUM_AND_ADD(Q,S)
      REAL Q(10,20,30),S
      SUM_AND_ADD = SUM(Q)+S
      RETURN
      END
C-- end of subprogram ----
------------------------------------------------------------
```

FIG. 24

FILE tiny1.o:

```
----------------------------------------------------
C-- main program ----
      PROGRAM SAMPL
      INTEGER N(100)
      REAL A(10,20,30),A2(10,20,30),B
      ...
      B = SUM_1(A)
      B = SUM_AND_ADD(A,B)
      WRITE(*,*) SUM_2(N(51:100))
      END
C-- end of main program ----
----------------------------------------------------
```

FILE tiny2.o:

```
----------------------------------------------------
C-- subprogram ----
      REAL FUNCTION SUM_AND_ADD(Q,S)
      REAL Q(10,20,30),S
      SUM_AND_ADD = SUM_1(Q)+S
      RETURN
      END
C-- end of subprogram ----
----------------------------------------------------
```

FILE onlines.o:

```
----------------------------------------------------
      REAL FUNCTION SUM_1(X)          }
      REAL X(1:10,1:20,1:30)          }
      SUM_1 = 0                       } PROCEDURE
      DO 999 I3 = 1, 30               } CODE A
      DO 999 I2 = 1, 20               }
      DO 999 I1 = 1, 10               }
       SUM_1 = SUM_1+X(I1,I2,I3)      }
  999 CONTINUE
      RETURN
      END

      INTEGER FUNCTION SUM_2(X)       }
      INTEGER X(51:100)               }
      SUM_2 = 0                       } PROCEDURE
      DO 999 I1 = 51, 100             } CODE B
       SUM_2 = SUM_2+X(I1)            }
  999 CONTINUE                        }
      RETURN                          }
      END                             }
----------------------------------------------------
```

FIG. 25

```
--------------------------------------------------------------------------
        SUBROUTINE SUBP(LEN)
        REAL,PARAMETER :: PAI=3.14159, R=100.0
        INTEGER LEN,M
        REAL :: S(2**LEN-1)
        ...
        M=PAI*(R*2)**2
        ...
        END SUBROUTINE
--------------------------------------------------------------------------
```

FIG. 26A

```
--------------------------------------------------------------------------
        SUBROUTINE SUBP(LEN)
        REAL,PARAMETER :: PAI=3.14159, R=100.0
        INTEGER LEN,M
        REAL :: S(POW_SUBP_1(2,LEN)-1)
        ...
        M=PAI*POW_SUBP_2((R*2),2)
        ...
        END SUBROUTINE
--------------------------------------------------------------------------
```

OBJECT PROGRAM

```
--------------------------------------------------------------------------
        FUNCTION POW_SUBP_1(A,N) RESULT(R)
        INTEGER A,R
        INTEGER N

        SELECT CASE (N)
        CASE (0)
          R=1
        CASE (1)
          R=A
        CASE (2)
          R=A*A
        CASE (3)
          R=A*A*A
        CASE DEFAULT
          R=A**N
        END SELECT
        RETURN
        END FUNCTION
--------------------------------------------------------------------------
```

ONLINE CODE A

```
--------------------------------------------------------------------------
        FUNCTION POW_SUBP_2(A,N) RESULT(R)
        REAL A,R
        INTEGER N

        R=A*A
        RETURN
        END FUNCTION
--------------------------------------------------------------------------
```

ONLINE CODE B

FIG. 26B

FIG. 27A

```
FUNCTION name(A,N) RESULT(R)
arg-type A,R
INTEGER N

R=1
RETURN
END FUNCTION
```

FIG. 27B

```
FUNCTION name(A,N) RESULT(R)
arg-type A,R
INTEGER N

R=A
RETURN
END FUNCTION
```

FIG. 27C

```
FUNCTION name(A,N) RESULT(R)
arg-type A,R
INTEGER N

R=A*A
RETURN
END FUNCTION
```

FIG. 27D

```
FUNCTION name(A,N) RESULT(R)
arg-type A,R
INTEGER N

R=A*A*A
RETURN
END FUNCTION
```

```
---------------------------------
FUNCTION name(A,N) RESULT(R)
arg-type A,R
INTEGER N

R=A**N
RETURN
END FUNCTION
---------------------------------
```

FIG. 28A

```
---------------------------------
FUNCTION name(A,N) RESULT(R)
arg-type A,R
INTEGER N

SELECT CASE (N)
CASE (0)
  R=1
CASE (1)
  R=A
CASE (2)
  R=A*A
CASE (3)
  R=A*A*A
CASE DEFAULT
  R=A**N
END SELECT
RETURN
END FUNCTION
---------------------------------
```

FIG. 28B

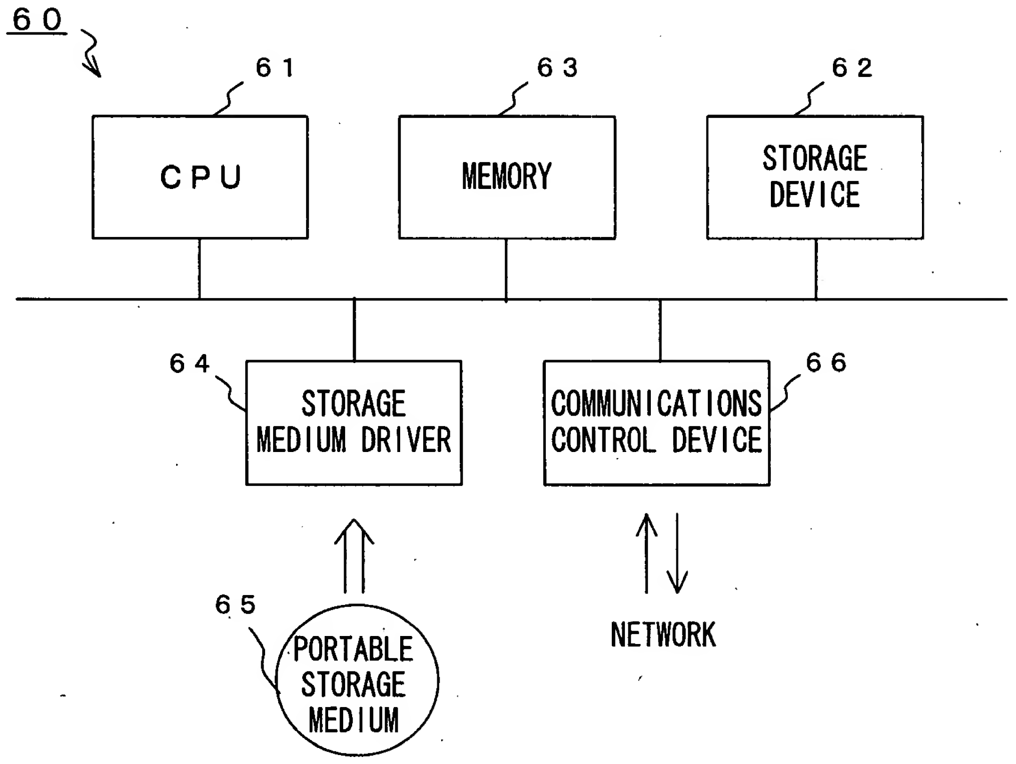

60
61
CPU
63
MEMORY
62
STORAGE DEVICE
64
STORAGE MEDIUM DRIVER
66
COMMUNICATIONS CONTROL DEVICE
65
PORTABLE STORAGE MEDIUM
NETWORK

FIG. 29

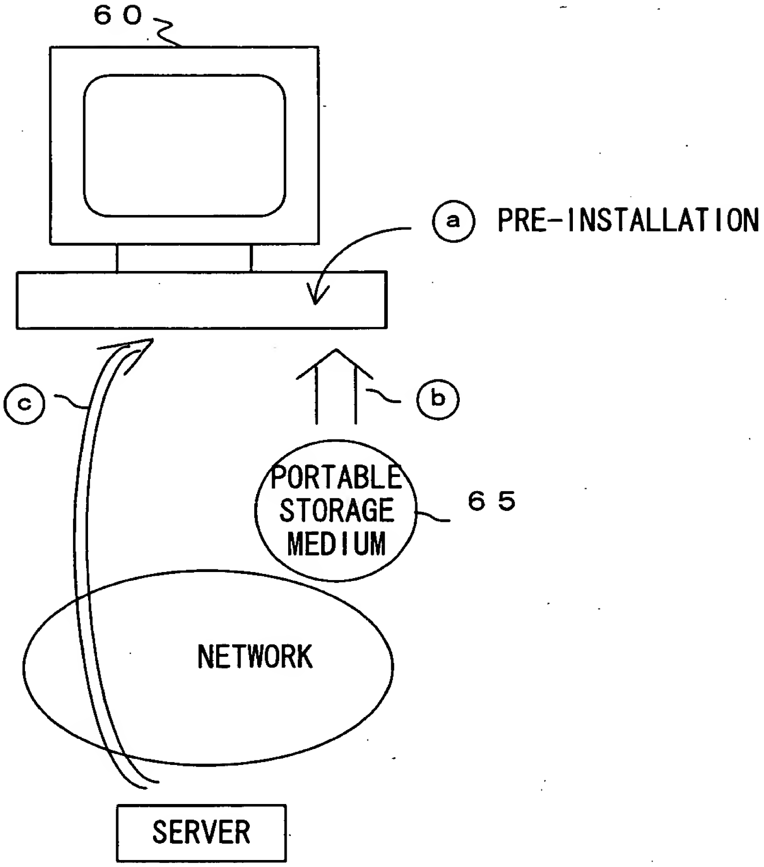

60
a
PRE-INSTALLATION
c
b
PORTABLE STORAGE MEDIUM
65
NETWORK
SERVER

FIG. 30